Diversity-driven Widening of Hierarchical Agglomerative Clustering

Alexander Fillbrunn and Michael R. Berthold

Chair for Bioinformatics and Information Mining
Dept. of CIS and Graduate School Chemical Biology (KoRS-CB)
University of Konstanz, 78457 Konstanz, Germany
firstname.lastname@uni-konstanz.de

Abstract. In this paper we show that diversity-driven widening, the parallel exploration of the model space with focus on developing diverse models, can improve hierarchical agglomerative clustering. Depending on the selected linkage method, the model that is found through the widened search achieves a better silhouette coefficient than its sequentially built counterpart.

1 Introduction

With the rise of multi-processor computer systems and multi-machine clusters, great efforts have been made to adapt machine learning to the changing paradigm of scaling hardware horizontally instead of vertically. Many traditional learning algorithms have been revised to run in a parallelized environment (eg. decision trees [18], neural networks [19] and SVMs [5]). These algorithms mostly focus on making the model building faster, but produce the same models as the non-parallel algorithms. Another approach that focuses on leveraging parallel computing resources to improve models generated by a data mining algorithm, rather than speeding up the computation, has been proposed in [1]. The technique has already been shown to work well for the set covering problem and *KRIMP* [17].

In this paper we describe a widened algorithm for hierarchical agglomerative clustering [6]. Parallel versions of this algorithm have been described in [14], however the focus there is again on acceleration rather than improving the model. Our preliminary results indicate that building multiple, diverse clustering models in parallel can improve the quality of the clustering for different quality metrics.

2 Widening

The widening technique for algorithms has first been described in [1]. It discusses an approach that focuses on leveraging parallel computing resources to improve models generated by a data mining algorithm, rather than speeding up the computation. Instead of greedily traversing the model space in search of a model that is just good enough, widening seeks to explore the space of all possible models in parallel, focusing on a certain number of best models at a time, iteratively refining them and selecting the best models again. Formalized, the standard way of searching the model space can be written as:

$$m' = s(r(m)) \tag{1}$$

where m is the current model and m' is the next model in the greedy search step. The function $r(\cdot)$ is the refinement of a model and $s(\cdot)$ the selection of the best model. The greedy search of the model space is therefore only a sequence of refinement and selection steps which terminates when a good enough model has been found. Widening, on the other hand, can be described using the following formula:

$$\{m'_1, \ldots, m'_{k'}\} = s(\{r(m_1), \ldots, r(m_k)\}) \ . \tag{2}$$

In a widened algorithm we do not deal with a single model, but with sets of models. The refinement operation produces multiple refinements from a single model and the selection filters them in order to return a set of best k' models. It can therefore be seen as a beam search through the model space. To avoid the selection operation choosing very similar models and not converging to a single solution or multiple very similar solutions, it is beneficial to enforce diversity within the selected models. Techniques for diversity-driven widening are discussed in [7]. One of the proposed methods is *Diverse Top-k Widening*, which makes use of a fixed diversity threshold θ that governs how similar the selected models are allowed to be, given a distance function δ.

3 Related Work

Since this paper focuses on widening a clustering algorithm, we focus here on work related to diversity-focused clustering and refer the reader to [1] and [7] for research into the general notion of enforcing diversity in model learning.

An approach that concentrates on diversity in clustering models is described in [2]. Here multiple diverse k-means clusterings are created in order to let the user choose the most applicable. Instead of selecting diverse clusterings after overproduction, the paper proposes a method whereby diversity is generated by running the k-means algorithm multiple times with different random initializations and random feature weighting. The large number of clusterings is then clustered at a meta level to present the user with a reasonable number of diverse models. The rationale here is that there are different clusterings for different purposes and the user ultimately knows best which one to choose. This, of course, is only useful for data sets with a low dimensionality.

Another paper that deals with finding better clustering results is [11]. Here the hierarchical clustering problem is solved using a genetic algorithm that tries to optimize the L_2 norm between an ultrametric distance matrix associated with the hierarchical classification and the proximity matrix of the dataset.

4 Widened Hierachical Agglomerative Clustering

In this paper we describe the widening of hierarchical agglomerative clustering. This bottom-up algorithm starts with every data point being a single cluster and subsequently merges the two clusters that are closest to each other. Apart from the distance function used to build the initial distance matrix, there are several possible linkage criteria for calculating the distance between newly formed clusters. Commonly used ones are:

UPGMA The *Unweighted Pair Group Method with Arithmetic Mean* calculates the distance between two merged clusters A and B and another cluster C as the mean of the distance between A and C and between B and C.

Complete linkage This method defines the distance of two clusters as the distance between those two data points (one from each cluster) that are farthest away from one another.

Single linkage Contrary to complete linkage, here the distance of two clusters is the distance between those two data points that are closest to each other.

Centroid linkage In this linkage method the distance between two clusters is the distance of their respective centroids.

Median linkage Here the distance between two clusters is the Euclidean distance between their *weighted* centroids.

Centroid and median linkage are notable because they do not lead to a monotone distance measure. The resulting clustering dendrograms can have inversions because the similarity between two clusters increases through a merge of one of them with another cluster. Even though this makes the dendrogram harder to interpret, the linkage criterion is often used because the similarity of two centroids is easy to understand.

The distances calculated with the above linkage methods are used to determine the two clusters to be merged in the next step. The algorithm continues to merge clusters until a predefined number of clusters is reached or until only one cluster is left. Because choosing the closest clusters to be merged is a local decision, what can occur is that the algorithm makes a merge that has a negative influence on future merges, where it may be forced to combine two clusters that do not fit together very well. Due to the greedy nature of the algorithm, widening can help to find better solutions by exploring a larger portion of the model space. While [7] also describes the notion of communication-free widening, we concentrate on the effect diversity has on the model building and allow the direct comparison of models in the selection step. Even though finding better models in the same amount of time is the eventual goal of widening, this paper does not take speed into account and focuses on creating better models than the sequential algorithm.

An efficient implementation of the hierarchical agglomerative clustering algorithm with a time complexity of $\Theta(N^2 \log N)$ can be found in [12]. It is based on priority queues that are used to quickly determine the closest neighbor of a given cluster. To achieve widening, we can make use of these queues by not only merging the closest pair, but also the second, third or hundredth closest and therefore generating many refinements from a single model. The number of refined models k_r in iteration i can be calculated as follows:

$$k_{r,i} = k * (N - i) \ . \tag{3}$$

Here N is the total number of data points to be clustered. In each iteration two clusters are merged into one, $(N - i)$ therefore denotes the number of clusters present in iteration i.

5 Achieving Diversity

The diversity of the models is enforced in the selection step, where we select k models from k_r refinements. Our goal is to select the most diverse and at the same time also best models to achieve both exploration and exploitation. This multi-objective problem is known as *Maximum-Score Diversity Selection* [13].

In the following chapters we introduce a distance metric for our models, which is based on the Robinson Foulds metric. Furthermore we describe how the quality of our models can be compared with a small extension of the standard heuristic of hierarchical agglomerative clustering.

5.1 Distance Metric for Hierarchical Clustering Models

To have a notion of (dis-)similarity for our models, we first need to define a distance metric. Since the clustering process merges clusters in a bottom-up fashion, the intermediate models are forests, where each tree is either a single data point or a cluster tree on a subset of all data points. Because the leaves of the trees in the forests are the original data points, all models have the same leafset. To calculate a distance between our models, we need a metric that can be applied to the forests. One such metric, even though originally used for calculating the distance between phylogenetic trees, is the Robinson Foulds metric [15]. This metric is based on the number of *bipartitions* shared by two trees. A bipartition is a split of the tree at an edge, so that the leaves are divided into two disjoint sets. Splits at edges that connect a leaf with the rest of the tree are called trivial bipartitions and are ignored for the calculation of the metric since they are present in every tree.

When $B(T)$ denotes the set of nontrivial bipartitions of a tree, the number of bipartitions found in a tree T_1 but not in another tree T_2 can be calculated as

$$|B(T_1) - B(T_2)| \ . \tag{4}$$

Using this the Robinson Foulds distance is defined as:

$$d_{RF}(T_1, T_2) = \frac{1}{2}(|B(T_1) - B(T_2)| + |B(T_2) - B(T_1)|) \ . \tag{5}$$

In order to apply the distance metric to our forests, we define the set of bipartitions for a forest F to be the union of bipartitions of its trees:

$$B(F) = \bigcap_{l=1}^{|F|} B(T_l) \ . \tag{6}$$

While the Robinson Foulds metric is originally devised for unrooted trees, these sets of bipartitions for forests allow us to calculate the distance between our models as well.

An efficient algorithm for computing the metric on trees has been given in [4]. As the first step of the algorithm for unrooted trees is to select one of the leaves as the root node, the fact that the Robinson Foulds distance was meant for unrooted trees is of no regard for our problem. Day's algorithm identifies nontrivial bipartitions by assigning intervals to each inner node of a tree. To obtain the set of intervals for a number of trees $T_1, \ldots, T_n$, we take T_1 and traverse it in a depth first fashion, labeling the leaves according to the order in which they are visited. This will be our reference labeling for the leaf nodes of all trees, which means that if leaf node A has label 1 in the reference labeling, it will have the same label in all of the trees under comparison. The labels are then used to calculate unique intervals for each inner node. An inner node's interval is the tupel of the largest and smallest label of all its descendant leaf nodes. A tree's interval set S_i is the set of tupels from all its inner nodes. Figure 1 shows two trees, where the left has been used to create the reference labeling of the leaves. The Robinson Foulds distance between those trees is 2, since their interval sets differ in two tupels.

In order to use Day's algorithm for our models, the leaf labels have to be assigned across multiple trees in a forest. For one model, its trees are ordered arbitrarily, then iterated and traversed depth first, labeling all the leaf nodes according to the order in which they are visited. Since all models have the same leaf nodes, the labels can be mapped to the nodes of the other forests as well. After obtaining a labeling for the leaves, the interval set for each tree is calculated as described above. To compare two forests F_1 and F_2, we compare the corresponding interval sets $B(F_1)$ and $B(F_2)$ by counting the intervals that occur in one set but not the other. Using this count we can create a $k_r \times k_r$ distance matrix D for all refined models.

5.2 Selecting diverse models

In the next step we need to select k models from the k_r refinements, choosing both good and diverse ones to find an even balance between global exploration and local exploitation of the model space. In the original algorithm for hierarchical agglomerative clustering the next model is the one where the two clusters that are closest to each other are merged. In the case of multiple models developed in parallel, we can improve this heuristic by using the aggregated merge distance as criterion. For each refined model m, the score $\phi_{m,i}$ in the current iteration i can be calculated as follows:

$$\phi_{m,i} = \sum_{j=1}^{i} d_{m,j} \tag{7}$$

where $d_{m,j}$ denotes the distances of the merged clusters in iteration j. The value $d_{m,j}$ depends on the distance metric used to build the initial distance matrix

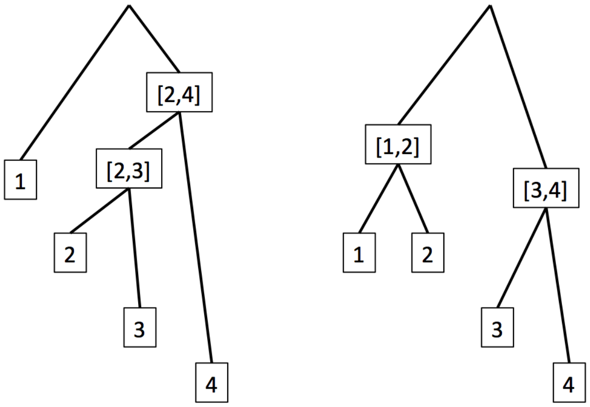


Fig. 1: Example of two trees and the corresponding intervals used by Day's algorithm to compute the Robinson Foulds distance.

and the linkage criterion that is used to calculate the distance between a merged cluster and all other clusters.

After each model has been assigned an associated score, we need to select models that are not only *good* according to our scoring function but also *diverse* according to our distance metric. While in [7] a Diverse Top-k approach is described, we propose another way of selecting diverse trees that does not rely on a diversity threshold θ. Because the trees get larger with each iteration, the distance between them also increases. A fixed threshold is therefore not suitable for this problem. Instead, diversity can be achieved by clustering the models into k clusters and picking the best model of each cluster for the output of the selection step (see Figure 2). Given the distance matrix D, we use k-medoid clustering [8] to split the set of models into groups and use $\phi_{m,i}$ to select the best model in each. The effect the model selection method has on diversity is demonstrated in Figure 3. Here 20 models were built in parallel on the seeds dataset from the UCI repository [10], using k-medoid clustering to enforce diversity. After 200 steps, when 10 clusters were left to be merged, the refinements of the current intermediate models were projected into 2D space using multidimensional scaling [9]. In Figure 3a the models that are chosen by the k-medoid selector for the next step are marked in red. Figure 3b shows which models would have been selected by a top-k selector. It can be seen that top-k focuses on a small area of the model space while models selected using k-medoid clustering are scattered across the whole space. The top-k approach also selects duplicates that occur in our models. The diversity enforcing clustering approach avoids this naturally as all equal models fall into the same cluster, but only one model is selected from each cluster.

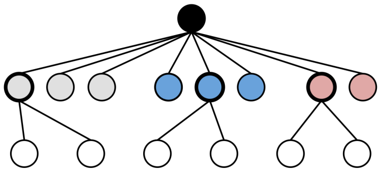


Fig. 2: The first step of widened model building using k-medoid with $k = 3$. Refined models are created from the initial model, then they are clustered into 3 groups and from each group the best model is used for creating the next generation of models.

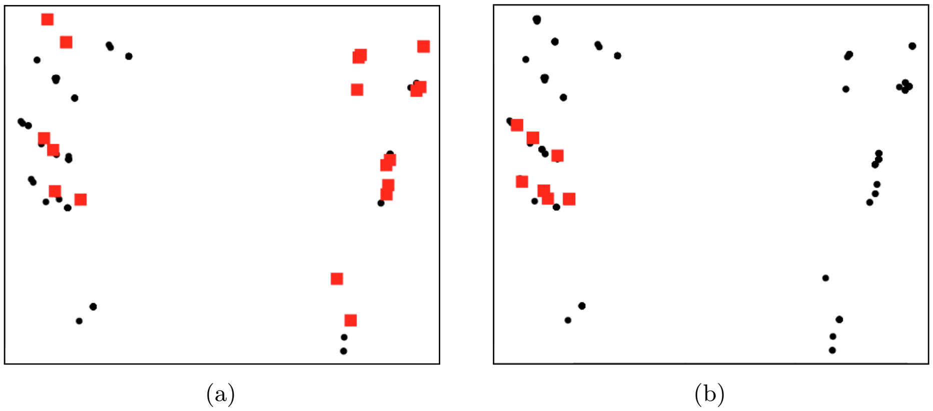


Fig. 3: Models projected with multidimensional scaling. In (a) the red squares mark the models selected using the k-medoid approach and in (b) the top-k models.

6 Evaluating Clustering Results

A commonly used quality measure for clustering results is the silhouette coefficient [16]. It is a number between -1 and 1, where values close to the lower bound are a sign of very bad clustering and numbers close to 1 mean that the found clusters are good. For an individual data point o belonging to cluster A the silhouette is defined as

$$s(o) = \frac{\mathrm{dist}(B, o) - \mathrm{dist}(A, o)}{\max\{\mathrm{dist}(A, o), \mathrm{dist}(B, o)\}} \tag{8}$$

where $\mathrm{dist}(A, o)$ is the average distance between o and all data points in A, and $\mathrm{dist}(B, o)$ is the distance between o and all data points in the next closest cluster B. The silhouette coefficient of a clustering result is the average $s(o)$ over all data points.

The Davies-Bouldin Index (DBI) [3] is another cluster evaluation measure that can be used to compare the quality of multiple clustering results. Like the silhouette coefficient it is an internal evaluation scheme, where only features of the dataset itself are taken into account. The index can be determined with the following formula:

$$DB = \frac{1}{N}\sum_{i=1}^{N} D_i \tag{9}$$

where D_i is defined as:

$$D_i = \max_{j \neq i} \frac{S_i + S_j}{dist(A_i, A_j)} \ , \tag{10}$$

with A_i being the centroid and S_i the scatter within cluster i:

$$S_i = \frac{1}{T_i}\sum_{d=1}^{T_i} ||X_d - A_i||_p \ . \tag{11}$$

Here T_i is the size of the cluster and X_d is a data point in the cluster. The Davies-Boulding-Index compares the within-cluster scatter to the between-cluster separation, represented by the distance between the corresponding centroids. A ratio close to zero means that the clusters are dense and well separated.

7 Preliminary Results

As our preliminary tests show, the best of multiple, built-in-parallel, diverse models can have both a better silhouette coefficient and Davies-Bouldin Index in comparison to the model found by the greedy, sequential algorithm. The effectiveness depends on the linkage method and the data set used. Tests have been carried out with the user knowledge modeling data set and the seeds data set from the UCI Machine Learning Repository. The data sets were chosen due to their suitability for clustering and their size. The desired number of clusters to be generated by the algorithms was set to 3 for the seeds data set and to 4 for the user knowledge modeling data set. We used the Euclidean distance as the distance measure for building the initial distance matrix for the data points and to calculate the between-cluster separation for the Davies-Bouldin Index.

In our tests clustering the seeds data set with median linkage shows promising results for the widened version of the algorithm. Figure 4a shows the silhouette coefficient of the best and worst of 10 widened models and the sequential algorithm's silhouette coefficient over the iterations of the algorithm. Here we can see that the widened algorithm generally produces a model with a better silhouette coefficient than the sequential algorithm.

Notable is the steep drop of the traditional algorithm's silhouette coefficient at 5 clusters (iteration 205), clearly visible in Figure 5a. Here it is forced to make a bad merge due to preceding greedy behavior. The best widened model also had a declining silhouette coefficient in previous iterations but has at that point already recovered with a silhouette coefficient of 0.389. If the data is clustered into 3 groups, the best widened model has a silhouette coefficient of 0.425. The sequential algorithm produces a model that has a silhouette coefficient of 0.264. Similar results can be achieved with centroid clustering. For average, complete and single linkage the silhouette coefficient could not be improved by widening.

The Davies-Bouldin Index, however, can be improved from 0.76 to 0.74 when the UPGMA linkage method is used. The best model obtained through widening the median linkage algorithm also achieves a lower DBI for 3 clusters. The best of the 10 widened models has a score of 0.65, the sequential algorithm achieves a DBI of 1.84.

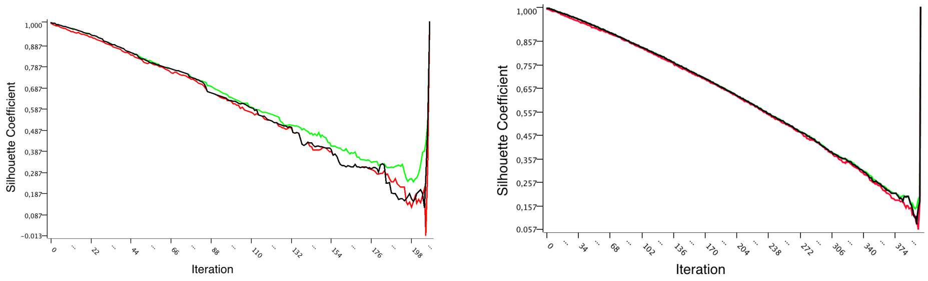
Similar results can be achieved when clustering the user knowledge modeling data set with complete linkage hierarchical clustering. Figure 4b depicts the silhouette coefficient for the best and worst of 10 widened models and the model generated by the sequential algorithm for each iteration of the algorithm. For 4 clusters the best widened model has a silhouette coefficient of 0.169, for the model generated by the sequential algorithm this value is 0.124. An interesting observation can be made in Figure 5b, where we see that the greedy algorithm's silhouette coefficient increases in iteration 384 but drops very low subsequently. The best widened model does not exhibit such extreme behavior. There the silhouette coefficient changes only slightly before dropping down to around 0.155.

The Davies-Bouldin Index also shows the improvement that is possible through widening. Clustering the user knowledge modeling data set with 4 desired clusters the widened algorithm produces a result with a DBI of 1.622 while the sequential algorithm achieves an index of 1.699. It is notable that the model with the lowest DBI does not also have the highest silhouette coefficient.

Note that the afore-mentioned widened algorithm's runtime is worse than the sequential algorithm's runtime, despite the possible parallelization of the refinement and selection processes. The reason for this increase in runtime is that calculating the pair-wise distance of many refined models for the matrix D is very time consuming, resulting in overhead for the selection step. This paper focuses on the role diversity plays in the intelligent search of the model space and performance improvements may be achieved by making the widened algorithm communication-less, avoiding the model-by-model comparisons altogether. This, however, is a topic of future research and not in the scope of this work. For an introduction to diverse communication-free widening we refer the reader to [7], where ideas for avoiding communication between parallel workers are described.

8 Conclusions and Future Work

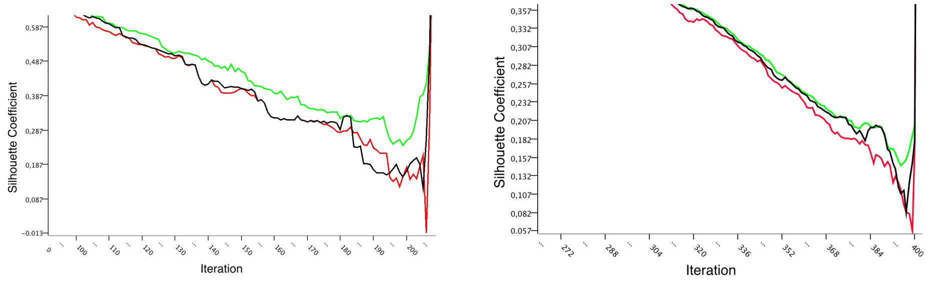
In this paper we have shown the application of widening to the hierarchical agglomerative clustering algorithm. The two main parts of widening are refinement and selection, for both of which we described implementations for hierarchical clustering. Creating refinements of a model utilizes information that is already present in the sequential algorithm, namely the priority queues that are maintained to keep track of the nearest neighbor of each cluster. For the selection of diverse and good models we described a method that groups models using



(a) Seeds data set

(b) User Knowledge Modeling data set

Fig. 4: The silhouette coefficient for intermediate models of the sequential algorithm (black) and the best (green) and worst (red) of the widened models for each iteration.



(a) Seeds data set

(b) User Knowledge Modeling data set

Fig. 5: The silhouette coefficients in the last steps of the sequential algorithm (black) and widened algorithm (best model: green, worst model: red).

k-medoid clustering and subsequently picks the best model from each group. We visualized how this approach covers the model space better than top-k, which focuses on a small area only. Our results on two public datasets indicate that the models obtained through widening can be better than the results of the sequential algorithm. This is the case for both the Davies-Bouldin Index and the silhouette coefficient, two widely used clustering evaluation metrics.

Future work includes the evaluation of other diversity facilitating methods such as p-dispersion-min-sum as well as making the algorithm communication-free. Removing communication between different branches of refined models would also increase the runtime performance of the algorithm, as less models would have to be compared to each other. This paper shows that spending parallel computing resources on exploring the model space can result in better models and widening the hierarchical agglomerative clustering algorithm is feasible when faster ways of enforcing diversity can be applied.

References

1. Zaenal Akbar, Violeta N. Ivanova, and Michael R. Berthold. Parallel data mining revisited. better, not faster. In *Proceedings of the 11th International Symposium on Intelligent Data Analysis*, pages 23–34, 2012.
2. Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. Meta clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 107–118. IEEE, 2006.
3. David L. Davies and Donald W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):224–227, April 1979.
4. William H.E. Day. Optimal algorithms for comparing trees with labeled leaves. *Journal of Classification*, 2(1):7–28, 1985.
5. Hans P Graf, Eric Cosatto, Leon Bottou, Igor Dourdanovic, and Vladimir Vapnik. Parallel support vector machines: The cascade svm. In *Advances in neural information processing systems*, pages 521–528, 2004.
6. Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.
7. Violeta N. Ivanova and Michael R. Berthold. Diversity-driven widening. *Advances in Intelligent Data Analysis XII*, 2013.
8. L. Kaufman and P. Rousseeuw. *Clustering by Means of Medoids.* Reports of the Faculty of Mathematics and Informatics. Faculty of Mathematics and Informatics, 1987.
9. Joseph B. Kruskal and Myron Wish. *Multidimensional scaling*, volume 11. Sage, 1978.
10. M. Lichman. UCI machine learning repository, 2013.
11. José Antonio Lozano and Pedro Larranaga. Applying genetic algorithms to search for the best hierarchical clustering of a dataset. *Pattern Recognition Letters*, 20(9):911–918, 1999.
12. Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
13. Thorsten Meinl. *Maximum-Score Diversity Selection.* PhD thesis, University of Konstanz, July 2010.

14. Clark F. Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21(8):1313 – 1325, 1995.
15. D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(12):131 – 147, 1981.
16. Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(0):53 – 65, 1987.
17. Oliver Sampson and Michael R Berthold. Widened krimp: Better performance through diverse parallelism. In *Advances in Intelligent Data Analysis XIII*, pages 276–285. Springer, 2014.
18. Anurag Srivastava, Eui-Hong Han, Vipin Kumar, and Vineet Singh. Parallel formulations of decision-tree classification algorithms. In Yike Guo and Robert Grossman, editors, *High Performance Data Mining*, pages 237–261. Springer US, 2002.
19. N. Sundararajan and P. Saratchandran. *Parallel Architectures for Artificial Neural Networks: Paradigms and Implementations.* IEEE Computer Society Press, Los Alamitos, CA, USA, 1st edition, 1998.